



Universidad
Carlos III de Madrid

TRABAJO FIN DE GRADO

Desarrollo de una aplicación móvil dentro del campo del m-health

Autor: Manuel Estada Calvente

Tutor: David Griol Barres

Grado: Doble Grado en Ingeniería Informática y Administración y
Dirección de Empresas.

Madrid, junio de 2017

Resumen

En el presente Proyecto de Final de Carrera se ha desarrollado una aplicación para Android consistente en un gestor de residencias para dar soporte a las residencias del Grupo Eulen. La aplicación permite consultar diversa información acerca de los centros, como puede ser información de localización y contacto de los centros, servicios y programas que ofrecen, y detalles de los pacientes que están en la base de datos de la empresa.

La aplicación permite al usuario introducir cadenas de texto mediante la voz o mediante el teclado de la pantalla. Esto favorece la interacción con la aplicación, debido a que le permite al usuario utilizar la aplicación en momentos en los que la interfaz tradicional impide o dificulta el uso. Para desarrollar esta funcionalidad, se ha realizado un estudio de los sistemas de dialogo, así como la forma de integrar estos mismos en Android.

Para definir las diferentes funcionalidades de la aplicación, se ha realizado un estudio previo del mercado. Se han encontrado diferentes aplicaciones para dispositivos Android pertenecientes a diferentes residencias en España de las cuales se han cogido ideas y se han mejorado funcionalidades implementadas en ellas.

La aplicación utiliza además otras tecnologías, como un gestor de bases de datos SQLite que se detalla en la memoria, con las interfaces y clases que ofrece para operar con los datos almacenados en las tablas de información acerca de los pacientes y las residencias tomadas como ejemplo.

Palabras clave: sistemas de diálogo, interacción oral, residencias de mayores, aplicación, Android.

Summary of the Bachelor's Project

Introduction

Thanks to the fast-growing technological advance in the last years, the way in which people communicate and obtain information has changed. One of the keys of this development is the emergence of smartphones. In 2015, it was recorded in Spain two important facts. The first of them is that six individuals out of ten aged 15 or older used a smartphone and the other one is that the mobile phone has become the main Internet access device (76.4%).

One of the most proliferated sectors in recent times regarding ICT (Information and Communication Technologies) is the health sector, which has been digitized giving rise to the appearance of the concept e-health. This concept refers to the practices related to the health care aided by communication systems and electronic processes. There are different systems and services, but in our case, we are going to focus on m-health ("mobile-health") services and systems. These services are characterized by being guided using smartphones to capture, analyze, process and transmit information on the health field through different systems.

Focusing on the issue of health through mobile devices we can see that there is a generational gap, since older people have more problems to get on in this area and get used to the changes that have raised up in recent decades. The idea for this project came as a response to the ENCAGE-CM program (<http://encage-cm.es/>), a program of R&D activities on active ageing, quality of life and gender, focused on Comunidad de Madrid. This program improves the standard of living of the elderly people. Specifically, this work expects to develop an application destined to improve the quality of life of people who are patients of nursing homes.

This application contains personal information, with which you must be very careful. Therefore, patient data will be stored and handled confidentially. The application must enforce the current legal regulations of the Data Protection Law and inform the Spanish Data Protection Agency of the data protection files used, indicating the purpose and the Treatment that it will be given to them.

It will also be necessary to protect the information stored in the databases through security measures and the keys will be encrypted to make it difficult to access them. This information about data protection must be present in the application in a place where the user can easily access. Therefore, in order to a user to be able to download the application, his consent must be requested and he must indicate that he "has read and accepted the terms and conditions of use".

It is also advisable to legally protect the application and this is possible in two ways. The first, and the one that we will consider, is through the "Intellectual Property Law", by which the application is considered a computer program. According to this law, the person who has the rights of the application is the person who created it or the legal entity if it was the result of the business activity. To protect the application, it must be registered in the Property Registry, providing a memory and the programming code used for the development.

Before publishing the application in the application stores you must create a developer account on the desired platform, in our case it will be Google Play. Once this is done you can upload the application with all the information about it and with screenshots of it.

Goals and objectives

The main goal of this project is to develop an Android mobile application that is located within the framework of m-health and aimed for improve and facilitate the management of patients in the residences, consequently it will have a positive impact on the patients because they will be better organized.

Within the definition of active aging we find another objective: "to optimize opportunities for physical, social and mental well-being throughout life with the aim of expanding healthy life expectancy, productivity and quality of life in old age".

As a practical example to develop this document, the idea is to implement the application in one of the nursing homes of the Eulen Group, located in the Community of Madrid. One of the features it offers is that users can interact with the application both screen or voice, and that users can be or the relatives of the patients or the workers of the residence. Therefore, the goal is to make a scalable application that can be adapted to other residences with ease.

One last objective that could mark as one of the most important is to learn to program in Android and learn to develop a mobile application, because throughout the degree we did not have the opportunity to study anything related to this and for me it seems to be a very important knowledge for the future due to the growth that the development of mobile applications is having.

In the present Project, it has been developed an application for Android consisting of a manager of residences to give support to the residences of the Eulen Group. The application allows you to consult various information about the

centers, such as location and contact information of the centers, services and programs they offer, and patient details that are in the company database.

The application allows the user to enter text strings using the voice or keyboard of the screen. This favors interaction with the application, because it allows the user to use the application at a time when the traditional interface hamper or makes difficult the usability. To develop this functionality, a study of the systems of dialogue as well as the way to integrate these same ones in Android have been realized.

In order to define the different functionalities of the application, a previous study of the market has been carried out. Different applications have been found for Android devices belonging to different nursing homes in Spain from which ideas have been taken and the functionalities implemented in them have been improved.

The application also uses other technologies, such as a SQLite database manager that is detailed in memory, with the interfaces and classes that it offers to operate with the data stored in the tables of information about patients and residences taken as an example.

The different functionalities that the application addresses throughout its different modules are:

- Search of the different centers belonging to the Eulen Group, as well as their contact and location data.

- Patient search engine, which allows users to search and access their data. The data of a particular patient can only be consulted by the relatives and staff of the center with permits. This will require a unique and non-transferable password to access a patient's profile.
- Profile of a patient, where you can find the medical data, the personal data of the employee who takes care of him or send a message to the center about the patient.
- Services and programs, which shows the different offer that the centers have, as well as the activities that are organized and offered.

Results

In order to evaluate the application, a survey was completed by 13 people, who used the application freely, choosing where to navigate and accessing the modules they wanted to test the application in full. The survey was generally carried out by family and friends, because it was not possible to go to any center to test the application. Responses have been unbiased and users have given tips for improve the application.

The results of the surveys were obtained thanks to the Google Forms tool used to design it, since it offered a section where the different answers obtained both individually and in a general way were summarized.

Once the results of the survey were obtained, they were analyzed in order to show conclusions about them.

In general, the tactile handling of the application and its navigability have been simple and positive. In any case a problem was caused to a user because there is no negative answer about these questions. The application has also a good interaction speed.

The quality of the information obtained is quite good, in any case this question has received a negative score. If it is true that only the 50% of the survey respondents gave the highest value to this question (5), so we can work out that the information offered by the application requires an improvement.

Voice interaction is a positively valued aspect, it was easy to know when and how to use it. Also is important the number of people who have not used it, so maybe it will be necessary to implement the voice interaction in more places in the application.

Of all users surveyed, only one error was found. The error was that some data of the patient profile was not showed when you go back to that screen.

Most of the users said that it is a useful application, a very positive result. Only a very small percentage regards the application as normal. Few users have rated the application as indispensable, so we can say that it needs improvements to increase this percentage.

Conclusions

The application developed consists of a manager of nursing homes that gives support to the residences of the Eulen Group. The application allows you to consult various information about the centers, such as location and contact information for the centers, services and programs they offer, and patient details that are in the company database. These functionalities cover the proposals that were determined at the time of the planning and design of the application.

The application has a multimodal system, in which the user has the possibility to use as input method both the voice and the keyboard. This feature has been implemented for a very small set of functions, but the application did not require it in any other section. It was interesting to apply the voice control system to the application because it provides user facilities for handling any application, since, as explained in Chapter 2, it can be used in multiple situations and without the need of access to Internet.

All the data that can be consulted in the application is stored in a SQLite database, since this tool is incorporated in Android and contains all the necessary functions for the storage and handling of data. This database manager does not need previous configuration, so to use it was only necessary to know well the interfaces and classes offered by the Android API to develop in the management of the tool.

Thanks to the relational design made based on data prior to its implementation, it has been possible to more easily adapt the necessary data of the application and define the queries on it so that they were optimal. As the application did not require a large amount of data, this manager has been sufficient to store the four tables

with their different data. In order for the operations on the database to not harm the operation or speed of the application, they have been executed in a different thread of the application as an asynchronous task.

Thanks to the development of the application it has been possible to acquire a lot of knowledge about the architecture and characteristics of the Android operating system, which before was a totally unknown world. In recent times, the topic of developing applications is booming, so it has not been very complicated to find help on the internet to develop and advance in the implementation of the application.

Apart from that, Android has public and free information of all the available packages, their interfaces and classes, as well as examples on how to use them. The development is done using Java code and XML, and that has influenced positively since you can find a large amount of code examples on the Internet. Also note that the programming language Java is one of the most studied in the degree, so this has not been a problem for programming in Android Studio.

Based on the evaluation obtained by different users, it concludes with a positive appraisal of the application. Users see it as usable, intuitive and easy to use. It is valued as a useful application, so that its implementation in the centers can have as a response that the objectives set at the beginning of the project are met. The main objective of the implementation of the application is to optimize the standard of living in old age, a concept known as active aging.

Then, and to conclude, it is important to detail the aspects to work in the future to improve the performance of the application developed. This includes improvements to the implemented modules, as well as new functionalities that can be incorporated to improve user satisfaction.

In the module of the residences, more specifically in the detail of them, add a link with Google Maps that allows the application to show the location of the residence, as well as a route from the selected location.

In the beginning, the application was designed to facilitate mobility inside the residence. This was not possible since it required an authorization that never arrived to be able to visit a residence and obtain the plans of it. As future work, the possibility of incorporating internal plans of residences is proposed. This will allow patients to become users of the application, allowing them to move around the residence more easily. The idea was that the application could guide the user through the residence by voice to further improve functionality.

Within the details of the residence, incorporate a button that allows the user to contact the residence instantly. Using the button would open the application of the phone in the case of pressing the phone or the application of mail in the case of press the email button. This would also be useful in the details of the worker when contacting him by telephone.

Add information to the data of the center, such as capacity, types of rooms or more specific activities that it will be taking place in it.

In the section of contacting the worker, establish a chat instead of a text box in order to maintain a conversation instead of sending a single message.

Índice general

1. INTRODUCCIÓN	17
1.1 Motivación.....	17
1.2 Objetivos.....	18
1.3 Estructura de la memoria.....	18
1.4 Proceso de desarrollo.....	19
1.5 Recursos utilizados	21
1.6 Marco regulador	22
1.7 Impacto socio-económico.....	23
2. ESTADO DEL ARTE.....	25
2.1 Sistemas de diálogo	25
2.2 Sistemas de reconocimiento de voz en Android.....	27
2.2.1 Integración de reconocimiento de voz en la aplicación	29
2.3 Sistemas de texto a voz en Android.....	30
2.3.1 Integración de síntesis de voz en la aplicación	31
2.4 Aplicaciones sobre residencias	32
3. DESCRIPCIÓN GENERAL DEL SISTEMA.....	34
3.1 Presentación del sistema.....	34
3.2 Tecnologías utilizadas	35
3.2.1 Base de datos.....	35
3.3 Módulos del sistema	41
4. DESCRIPCIÓN DETALLADA DE LOS COMPONENTES DEL SISTEMA.....	42
4.1 Módulo “Inicio”.....	42
4.2 Módulo “Servicios”	43
4.3 Módulo “Centros”	44
4.3.1 Submódulo “Detalles de la residencia”	45
4.4 Módulo “Pacientes”	46
4.4.1 Submódulo “Detalles del paciente”	48
4.4.2 Submódulo “Login”	49
4.4.3 Submódulo “Perfil del paciente”	50
5. EVALUACIÓN DE LA APLICACIÓN.....	53
5.1 Método de evaluación.....	53
5.2 Resultados de la evaluación.....	55
6. CONCLUSIONES Y TRABAJO FUTURO	59
6.1 Conclusiones.....	59
6.2 Trabajo futuro	60

Índice de figuras

Figura 1. Tabla detallada de la planificación.....	20
Figura 2. Diagrama de Gantt	21
Figura 3. Procesamiento del habla de los sistemas de diálogo.	26
Figura 4. Ajustes del dictado por voz de Google	27
Figura 5. Aplicación de búsqueda por voz de Google.....	28
Figura 6. Ajustes de síntesis de voz.....	30
Figura 7. Aplicación Residencia Palacios	32
Figura 8. Aplicación Residencia de la Soledad.	33
Figura 9. Aplicación Residencias Sanyres	33
Figura 10. Funcionalidades ofrecidas por la aplicación	35
Figura 11. Diagrama del modelo de base de datos	40
Figura 12. Módulo “Inicio” de la aplicación.	42
Figura 13. Pantalla 1 del módulo “Servicios”	43
Figura 14. Pantalla 2 del módulo “Servicios”	43
Figura 15. Pantalla principal del módulo “Centros”.....	44
Figura 16. Ejemplo de inserción en Base de Datos.	44
Figura 17. Submódulo “Detalles de la residencia”.....	45
Figura 18. Ejemplo de consulta en base de datos.	45
Figura 19. Pantalla inicial del módulo de pacientes.	46
Figura 20. Ejemplo de entrada por voz de la aplicación.	47
Figura 21. Ejemplo de búsqueda de un paciente.	47
Figura 22. Pantalla del submódulo “Detalles del paciente”.	48
Figura 23. Pantalla del submódulo “Login”.	49
Figura 24. Ejemplo de creación de la tabla Pacientes.	49
Figura 25. Ejemplo de contraseña incorrecta.	50
Figura 26. Pantalla principal del submódulo “Perfil del paciente”.	50
Figura 27. Pantalla de datos médicos del paciente.	51
Figura 28. Pantalla de datos personales del trabajador.....	52
Figura 29. Pantalla para contactar con el trabajador.	52
Figura 30. Encuesta de evaluación de la aplicación.	54
Figura 31. Resultados de la evaluación: pregunta 1.	55
Figura 32. Resultados de la evaluación: pregunta 2.	55
Figura 33. Resultados de la evaluación: pregunta 3.	56
Figura 34. Resultados de la evaluación: pregunta 4.	56
Figura 35. Resultados de la evaluación: pregunta 5.	56
Figura 36. Resultados de la evaluación: pregunta 6.	57
Figura 37. Resultados de la evaluación: pregunta 7.	57

Índice de tablas

Tabla 1. Coste de los recursos utilizados.....	23
Tabla 2. Previsión de ingresos año 1.	24
Tabla 3. Interfaz y clases del paquete android.speech.....	29
Tabla 4. Interfaces y clases del paquete android.speech.tts.....	31
Tabla 5. Resumen de comandos en lenguaje SQL.	36
Tabla 6. Interfaces y clases del paquete android.database.sqlite.....	37
Tabla 7. Métodos de la clase SQLiteOpenHelper.	38
Tabla 8. Métodos relevantes de la clase SQLiteDatabase	38
Tabla 9. Atributos de la tabla Pacientes	39
Tabla 10. Atributos de la tabla Datos	39
Tabla 11. Atributos de la tabla Trabajadores.....	40
Tabla 12. Atributos de la tabla Residencias	40

Capítulo 1

Introducción

Este capítulo está dedicado a la presentación del Trabajo de Fin de Grado, describiendo la motivación del mismo, los objetivos marcados para su desarrollo, la estructura del documento, la planificación realizada previa al desarrollo, los recursos utilizados, un estudio socio-económico del impacto que tendría la aplicación y el marco regulador que envuelve al desarrollo de la aplicación.

1.1 Motivación

Gracias al rápido avance tecnológico en los últimos años, ha cambiado la forma que tenemos las personas de comunicarnos y de obtener información. Una de las claves de esta evolución es la aparición de los teléfonos inteligentes (del inglés “*smartphone*”). En 2015 se registró en España que seis de cada diez individuos de 15 años o más utilizaban un *smartphone* y el teléfono móvil se ha convertido en el principal dispositivo de acceso a Internet (76,4%) [1].

Uno de los sectores que más está proliferando en los últimos tiempos en relación a las TIC (Tecnologías de la Información y Comunicación) es el sector sanitario, el cual se ha digitalizado dando lugar a la aparición del concepto *e-health* [2]. Este concepto se refiere a las prácticas relacionadas con el cuidado de la salud asistida por sistemas de comunicación y procesos electrónicos. Hay diferentes sistemas y servicios, pero en nuestro caso nos vamos a centrar en los servicios y sistemas *m-health*. Estos servicios se caracterizan por estar guiados por el uso de teléfonos inteligentes para capturar, analizar, procesar y transmitir información sobre el ámbito de la salud mediante diferentes sistemas [3].

La idea de este proyecto surgió como respuesta al programa ENCAGE-CM (<http://encage-cm.es/>), un programa de actividades de I+D sobre envejecimiento activo, calidad de vida y género, centrado en la Comunidad de Madrid. Este programa permite mejorar el nivel de vida de las personas mayores. Concretamente este trabajo pretende desarrollar una aplicación destinada a mejorar la calidad de vida de las personas que son pacientes de residencias para personas mayores.

1.2 Objetivos

El principal objetivo de este proyecto es el de desarrollar una aplicación móvil en Android situada dentro del marco del *m-health* y orientada a mejorar y facilitar la gestión de los pacientes de las residencias, por lo que indirectamente tendrá un impacto positivo en los pacientes de las mismas, que estarán mejor organizados.

Dentro de la definición de envejecimiento activo encontramos otro de los objetivos: “optimizar las oportunidades de bienestar físico, social y mental durante toda la vida con el objetivo de ampliar la esperanza de vida saludable, la productividad y la calidad de vida en la vejez” [\[4\]](#).

Como ejemplo práctico para desarrollar este documento, la idea es implementar la aplicación en una de las residencias de mayores del Grupo Eulen, ubicada en la Comunidad de Madrid. Una de las funcionalidades que ofrece es que los usuarios pueden interactuar con la aplicación tanto por pantalla como por voz, y que los usuarios puedan ser tanto los familiares de los pacientes como los trabajadores de la residencia. Por lo tanto, el objetivo es realizar una aplicación escalable, que pueda ser adaptada a otras residencias con facilidad.

Un último objetivo que podría marcar como uno de los más importantes es el de aprender a programar en Android y aprender a desarrollar una aplicación móvil, debido a que a lo largo de la carrera no hemos tenido la oportunidad de ver nada relacionado con esto y me parece que es un conocimiento muy importante para el futuro debido al crecimiento que está teniendo el desarrollo de aplicaciones móviles.

1.3 Estructura de la memoria

El documento está compuesto de ocho capítulos y a continuación se detalla la descripción de todos ellos:

- **Capítulo 1. Introducción.** En este capítulo se describe la motivación del trabajo, los objetivos planteados para su realización, la estructura de la misma, la planificación del desarrollo, los recursos utilizados, el presupuesto necesario para alcanzar los objetivos, el impacto socio-económico que tendría la aplicación a desarrollar y el marco regulador que la envuelve.

- **Capítulo 2. Estado del arte.** En este capítulo se tratan temas acerca de las tecnologías móviles, los sistemas utilizados por la aplicación como el control por voz y se hace un estudio de aplicaciones ya existentes en el ámbito de la salud y que estén relacionadas con las residencias de mayores.
- **Capítulo 3. Descripción del sistema.** Se describe la aplicación detallando las funcionalidades del sistema, explicando los componentes que la forman y las tecnologías utilizadas para su desarrollo.
- **Capítulo 4. Descripción detallada de los componentes del sistema.** En este capítulo se describen detalladamente todos los módulos de la aplicación y sus submódulos, añadiendo capturas de pantalla de la aplicación y explicando el funcionamiento de cada una de las mismas, así como detalles de implementación y desarrollo.
- **Capítulo 5. Evaluación de la aplicación.** El capítulo muestra una evaluación del funcionamiento de la aplicación mediante una encuesta con diferentes preguntas. Posteriormente se analizan los resultados obtenidos mediante gráficas.
- **Capítulo 6. Conclusiones y trabajo futuro.** En este apartado se presenta una visión personal del trabajo realizado, así como un planteamiento de mejoras futuras que se podrían realizar sobre la aplicación desarrollada.
- **Bibliografía.** En este apartado se citan las referencias bibliográficas utilizadas para la realización de este proyecto.
- **Glosario.** En este capítulo se definen los términos utilizados a lo largo de la memoria que pueden causar confusión a la hora de interpretarlos por parte del lector.

1.4 Proceso de desarrollo

En este apartado se describen las fases de desarrollo del proyecto. Cada fase está compuesta por diferentes procesos que se explican a continuación. El proceso de desarrollo se puede ver más claro en el diagrama de Gantt realizado con el programa Microsoft Project que se incluye después de la explicación.

Fase 1. Planificación: esta fase consiste en la búsqueda de información que va a ser necesaria para la realización del proyecto, así como el análisis y diseño de la aplicación. En esta fase también se estudia el sistema operativo Android, así como

el programa Android Studio utilizado para el desarrollo de la aplicación. También se evalúan los diferentes sistemas utilizados y las librerías implementadas para incorporar las diferentes funcionalidades definidas para la aplicación.

Fase 2. Desarrollo: en esta fase se definen de manera detallada las diferentes componentes que contiene la aplicación y se programa usando Android Studio las distintas funcionalidades mencionadas anteriormente.

Fase 3. Evaluación y Depuración: en esta parte se realizan pruebas de los diferentes componentes programados en la fase anterior y se arreglan los posibles fallos que se puedan encontrar para así mejorar el rendimiento de la aplicación a la hora de ser usada por un usuario real. También se realizará una evaluación de la aplicación.

Fase 4. Documentación: esta fase está compuesta por la redacción de la memoria y por la realización de la presentación final que se mostrara el día de la defensa del proyecto.





		Name	Duration	Start	Finish	Predecessors
1		☐ Fase 1. Planificación	41d	25/01/2017	22/03/2017	
2		Estudio de los sistemas de diálogo	10d	25/01/2017	07/02/2017	
3		Estudio de la plataforma Android Studio	10d	25/01/2017	07/02/2017	
4		Análisis de aplicaciones alternativas	2d	02/03/2017	03/03/2017	2,3
5		Definición de funcionalidades de la aplicación	5d	06/03/2017	10/03/2017	4
6		Estudio de las tecnologías utilizadas	8d	13/03/2017	22/03/2017	5
7		☐ Fase 2. Desarrollo	49d	23/03/2017	30/05/2017	6
8		Diseño detallado	14d	23/03/2017	11/04/2017	
9		Programación de la aplicación	35d	12/04/2017	30/05/2017	8
10		☐ Fase 3. Evaluación y depuración	8d	31/05/2017	09/06/2017	9
11		Pruebas	5d	31/05/2017	06/06/2017	
12		Evaluación	3d	07/06/2017	09/06/2017	11
13		☐ Fase 4. Documentación	86d?	06/03/2017	03/07/2017	
14		Redacción de la memoria	78d?	06/03/2017	21/06/2017	
15		Preparación de la presentación	6d?	26/06/2017	03/07/2017	

Figura 1. Tabla detallada de la planificación

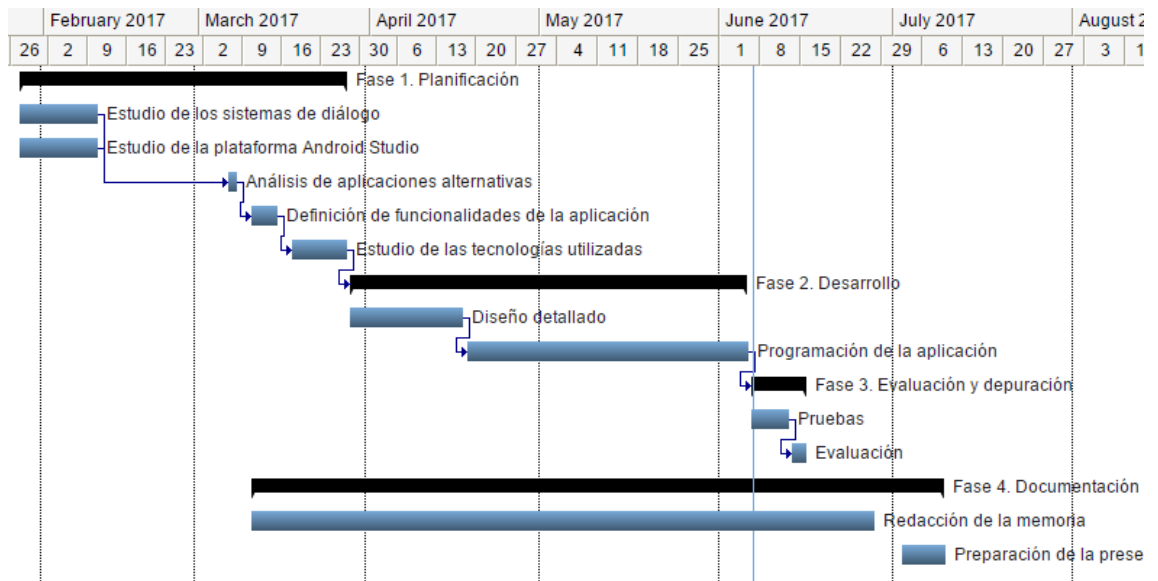


Figura 2. Diagrama de Gantt

1.5 Recursos utilizados

Durante el desarrollo de este proyecto se han utilizado diferentes herramientas, tanto hardware como software, que se detallan a continuación:

- Ordenador portátil Acer Aspire E 15
- Smartphone Doodgee Latte DG450
- Cable USB
- Android Studio 2.3.1
- Windows 10
- Android SDK
- JDK
- Microsoft Office 2016
- Microsoft Project 2010

1.6 Marco regulador

Esta aplicación contiene información personal, con la que se debe tener mucho cuidado. Por ello los datos de los pacientes se almacenarán y se manejarán de forma confidencial. La aplicación tiene que cumplir la normativa legal vigente de la Ley de Protección de Datos (LOPD 15/1999 de 13 de diciembre) e informar a la Agencia Española de Protección de Datos de los ficheros de protección de datos utilizados, indicando la finalidad y el tratamiento que se les va a dar.

También será necesario proteger la información almacenada en las bases de datos mediante medidas de seguridad y las claves estarán cifradas para dificultar el acceso a las mismas. Esta información debe estar presente en la aplicación en un lugar donde el usuario pueda acceder con facilidad. Por lo tanto, para que un usuario pueda descargarse la aplicación, se deberá pedir su consentimiento y deberá indicar que *“ha leído y aceptado los términos y condiciones de uso”*.

También es conveniente proteger legalmente la aplicación y esto es posible mediante dos formas. La primera, y la que tendremos en cuenta, es mediante la “Ley de Propiedad Intelectual”, por la cual se considera a la aplicación un programa de ordenador. Según esta ley, el que posee los derechos de la aplicación es la persona que la ha creado o la persona jurídica si ha sido resultado de la actividad empresarial. Para proteger la aplicación se debe registrar en el Registro de la Propiedad, entregando una memoria y el código de programación empleado para el desarrollo.

Antes de publicar la aplicación en las tiendas de aplicaciones se debe crear una cuenta de desarrollador en la plataforma deseada, en nuestro caso será Google Play. Una vez hecho esto se puede subir la aplicación con toda la información acerca de ella y con capturas de pantalla de la misma.

1.7 Impacto socio-económico

En la siguiente tabla se mostrarán los diferentes costes de los recursos, tanto software como hardware, necesarios para el desarrollo de este proyecto.

Recurso	Coste
Ordenador portátil Acer Aspire E-15	569 €
Smartphone Doodge Latte DG450	200 €
Cable USB	2 €
Android Studio 2.3.1	0 €
Windows 10	59 €
Android SDK (Software Development Kit)	0 €
JDK (Java Development Kit)	0 €
Microsoft Office 2016	79 €

Tabla 1. Coste de los recursos utilizados

Por lo tanto, la inversión inicial que se debería hacer para comenzar el desarrollo del proyecto es de 909 €.

Como punto de partida para analizar el impacto socio-económico de la aplicación hay que tener en cuenta que durante el pasado siglo España ha doblado la esperanza de vida y ha aumentado en un 24% el número de personas por encima de los 65 años. España cuenta con más de 7 millones de personas mayores, un porcentaje cerca del 20% con respecto al total de la población española (datos según por el INE). Esto tiene como consecuencia que haya más residencias y por lo tanto mayor número de clientes potenciales.

En nuestro caso, los potenciales usuarios serán en un principio las residencias del Grupo Eulen de la Comunidad de Madrid. Cuentan con 16 residencias/centros de día en la comunidad, pero se prevé en un futuro ampliar el mercado de referencia a las demás comunidades autónomas.

Al haber contactado con una de las residencias, la promoción de la aplicación es sencilla, puesto que, si esta tiene éxito una vez implantada en una residencia, se irá desarrollando la aplicación para el resto de residencias.

Los ingresos provendrán de la implantación y mantenimiento de la aplicación, debido a que la descarga de la misma será gratuita para los usuarios. La implantación de la misma tendrá un coste de 130 €. Por otro lado, el mantenimiento tendrá un coste de 150 €

mensual que se cobrará durante toda la vida útil de la aplicación. Por lo tanto, la previsión de ingresos en el primer año sería de 1930 €.

Esto sería teniendo en cuenta que la aplicación solo se implanta en una residencia. Si la aplicación tiene el éxito que se prevé, la previsión de ingresos se multiplicaría. En la siguiente tabla se refleja la previsión de ingresos que se tendría si la aplicación se implanta en tres residencias durante el primer año.

	1	2	3	4	5	6	7	8	9	10	11	12
IMPL.	130	0	0	130	0	0	0	0	0	130	0	0
MANT.	0	150	150	150	300	300	300	300	300	300	450	450
Total	130	150	150	280	300	300	300	300	300	430	450	450
Acumulado	130	280	430	710	1010	1310	1610	1910	2210	2640	3090	3540

Tabla 2. Previsión de ingresos año 1.

Capítulo 2

Estado del arte

Este capítulo tiene como finalidad estudiar las diferentes tecnologías utilizadas en el desarrollo de la aplicación a modo de poner en contexto este proyecto. Como parte central de esta aplicación, se deben identificar las diferentes posibilidades que ofrece Android a la hora de desarrollar aplicaciones que interactúan oralmente con el usuario. Se entrarán a explicar los diferentes sistemas de dialogo y los sistemas de reconocimiento de voz y de texto disponibles en Android. Por último, se estudiarán las diferentes aplicaciones que podemos encontrar en el campo de la salud y las aplicaciones dedicadas a residencias de mayores, relevante para el desarrollo de la aplicación de este proyecto.

2.1 Sistemas de diálogo

Los sistemas de dialogo o sistemas conversacionales (SLS, *Spoken Language Systems*) son una tecnología utilizada para facilitar la interacción mediante el habla de las personas con una máquina. El uso de la voz es una de las mejores maneras encontradas para establecer el contacto entre un usuario y un dispositivo electrónico [5]. Esta tecnología utilizada en aplicaciones móviles permite a una persona utilizar simplemente su voz para obtener información directa de un dispositivo o para controlar un determinado sistema, remplazando el uso de un teclado, de una pantalla o de un ratón, que son los medios convencionales que siempre se han utilizado [6].

Aun con el desarrollo de la tecnología en los últimos años, los sistemas de dialogo disponibles hoy en día tienen ciertas limitaciones, ya que no existe ninguno capaz de procesar con un cien por cien de precisión la variación del habla, por ejemplo, de la diferente fonética de personas de distinta procedencia geográfica o social, o las interferencias que ocurren por el entorno donde se utiliza el sistema, es decir, si se está utilizando en plena calle o dentro de un vehículo [7].

El reconocimiento del habla usa señales de voz en lugar de métodos tradicionales mencionados anteriormente. El proceso de este tipo de sistemas se puede ver en la siguiente figura [8], que muestra la arquitectura de un sistema de dialogo basado en el procesamiento del habla. Posteriormente se pasará a describir las diferentes acciones con las que cuenta el sistema.

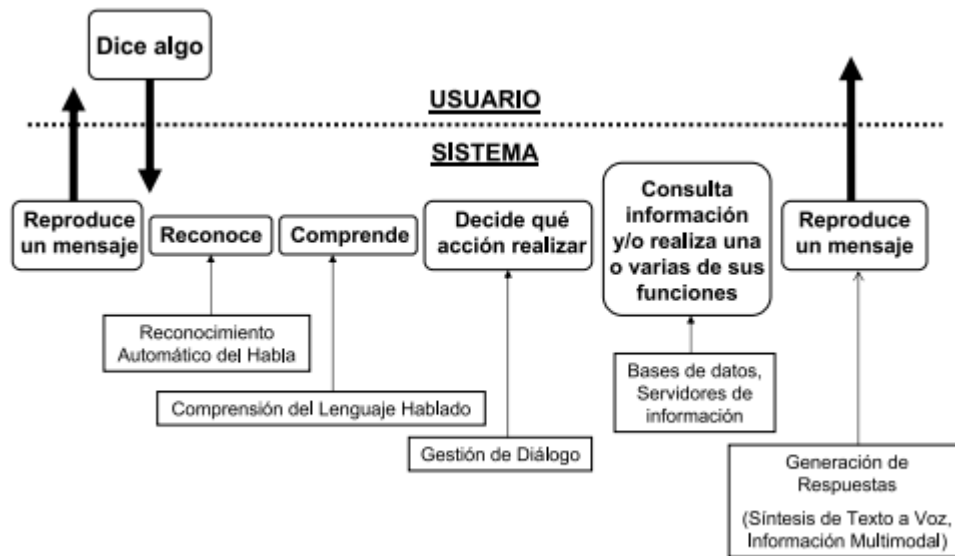


Figura 3. Procesamiento del habla de los sistemas de diálogo.

Este proceso empieza con una entrada oral proporcionada por parte del usuario, a partir de la cual entra a operar el sistema de diálogo, el cual realiza las siguientes funciones [9]:

- **Reconoce:** convertir la onda sonora en una representación escrita que pueda utilizar el sistema en los siguientes pasos (módulo de reconocimiento automático del habla).
- **Comprende:** el sistema debe extraer la información haciendo una interpretación semántica, recurriendo a técnicas de comprensión del lenguaje natural (módulo de comprensión del habla).
- **Decide que acción realizar** dependiendo de la petición realizada por el usuario (gestor de diálogo).
- **Consulta información y/o realiza una o varias de sus funciones:** en esta parte entran en juego las bases de datos, de las cuales el sistema debe obtener información para formular una frase gramaticalmente bien formada (módulo de consulta a la base de datos de la aplicación).
- **Reproduce un mensaje:** el sistema emite la respuesta que ha seleccionado (módulo de generación de respuestas y sintetizador de texto a voz).

2.2 Sistemas de reconocimiento de voz en Android

Desde que apareció el sistema operativo Android, Google ha proporcionado diferentes métodos de entrada para interactuar con los dispositivos con este sistema. En este apartado se va a tratar un método de entrada en concreto: la voz.

Fue a partir de la versión 2.1. de Android cuando Google integró el servicio de reconocimiento de voz en los dispositivos. A partir de ese momento, esta funcionalidad fue mejorando permitiendo al usuario utilizar la voz para interactuar con su teléfono con múltiples aplicaciones. Con la versión 4.0 se incorporó a la entrada por voz la capacidad para transcribir los resultados del reconocimiento a medida que el usuario iba hablando sin necesidad de esperar a que este hubiera terminado. En las últimas actualizaciones del sistema operativo podemos ver como se permite utilizar el reconocimiento por voz sin necesidad de disponer de una conexión a internet. También se introdujo la conversación bidireccional, es decir, una aplicación es capaz de formular una pregunta al usuario y así conseguir mejorar el uso de la misma.

La entrada por voz al teclado es posible gracias a un botón en forma de micrófono que se sitúa al lado de cualquier campo de texto. Para habilitar esta opción es necesario ir a *Ajustes>Idioma y entrada de texto>Teclado y métodos de entrada* y dentro de esa sección pulsar *Dictado por voz de Google*. Después tienes que acceder a las opciones del teclado de Android y seleccionar *Tecla de entrada de voz* y elegir “*Micrófono en teclado principal*”.

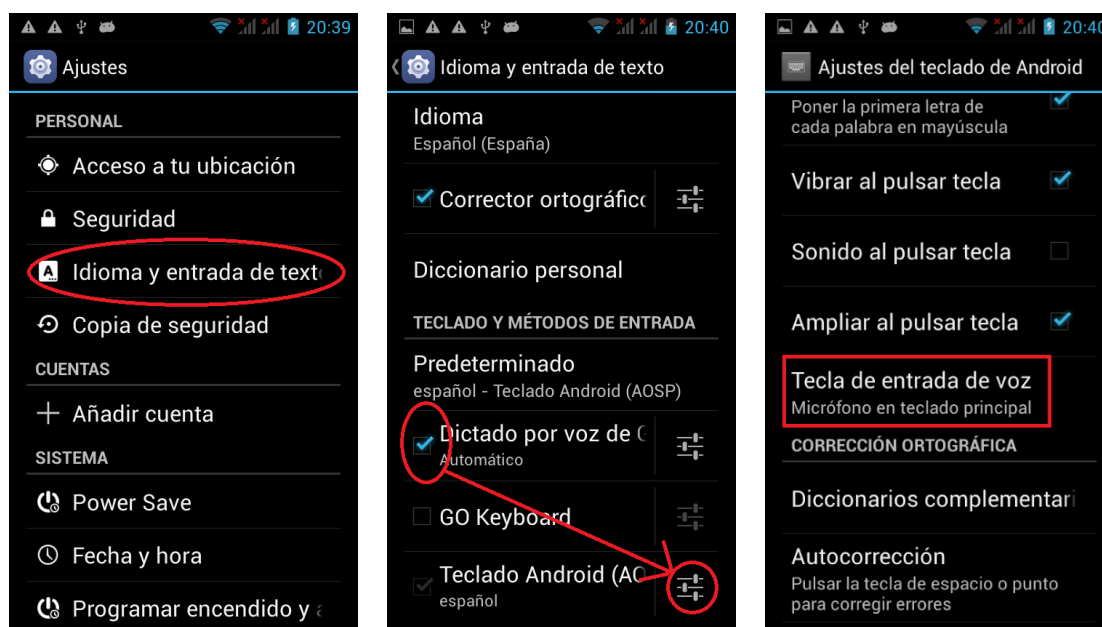


Figura 4. Ajustes del dictado por voz de Google

Para probar esta funcionalidad, la mayoría de los dispositivos Android cuentan con una aplicación de Google llamada “*Búsqueda por voz*”. Al abrirla nos aparece una interfaz muy sencilla con un campo de texto y un símbolo de un micrófono. Si pulsas el micrófono y hablas, el texto transcrito de las palabras aparece y aparecerán a continuación los resultados de tu búsqueda en los servidores de Google.

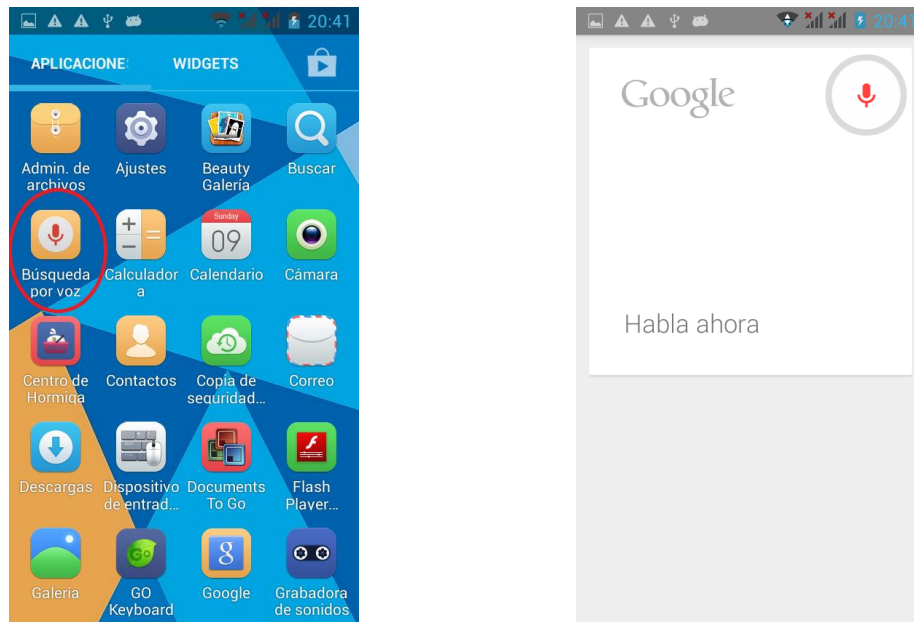


Figura 5. Aplicación de búsqueda por voz de Google

2.2.1 Integración de reconocimiento de voz en la aplicación

Mediante el SDK de Android, los desarrolladores tienen a su disposición diferentes maneras de introducir el reconocimiento automático del habla en sus aplicaciones. En este caso la aplicación va a hacer uso del paquete *android.speech*, el cual dispone de una variedad de interfaces y clases para integrar en una aplicación servicios de reconocimiento de voz ya creados o bien diseñar servicios nuevos. La siguiente tabla [10] resume las clases e interfaces que ofrece el paquete.

RecognitionListener	Recibe notificaciones del reconocedor de voz (SpeechRecognizer) cuando se ha dado algún evento relacionado con dicho reconocedor de voz.
----------------------------	--

(a) Interfaces

RecognitionService	Sirve de base para implementar un servicio de reconocimiento de voz.
RecognitionService.Callback	Recibe las notificaciones del servicio de reconocimiento de voz y se las envía al usuario.
RecognizerIntent	Define las constantes necesarias para integrar el reconocimiento de voz iniciado desde un <i>intent</i> .
RecognizerResultsIntent	Define las constantes asociadas a intents que se encargan de mostrar los resultados del reconocimiento de voz.
SpeechRecognizer	Clase que proporciona acceso al servicio de reconocimiento de voz, el cual permite tener acceso al reconocedor de voz.

(b) Clases

Tabla 3. Interfaz y clases del paquete *android.speech*

Para integrar el reconocimiento de voz en la aplicación, la opción más sencilla y empleada en esta aplicación es la de lanzar un intent de tipo *android.speech.RecognizerIntent*, el cual genera la activación del grabador de voz de Android solicitando al usuario que proporcione la entrada por voz. El audio grabado se envía a los servidores de Google para que sea procesado. Por último, los resultados del reconocimiento de voz son enviados a la actividad que lanzó el intent inicial.

2.3 Sistemas de texto a voz en Android

Un sistema de síntesis de texto a voz es una herramienta que sirve para producir una señal sonora en lenguaje natural que represente a un texto escrito. El objetivo principal que tiene es el de reproducir el texto mediante la voz, permitiendo al usuario interactuar con la aplicación sin necesidad de leer la pantalla del dispositivo.

Desde el sistema operativo Android 1.6 se incorporó un motor de síntesis de voz denominado Pico TTS desarrollado por la empresa SVOX y Google. Este sistema permite integrar este mecanismo en cualquier aplicación, por lo que las aplicaciones son capaces de producir voz a partir de una cadena de texto.

Para habilitar esta funcionalidad se debe ir a *Ajustes>Idioma y entrada de texto>Síntesis de voz*. Podemos encontrar ahí los motores (en nuestro caso sólo el predeterminado) y configurar el idioma, la velocidad de la voz y escuchar un ejemplo de síntesis de voz con el motor seleccionado.

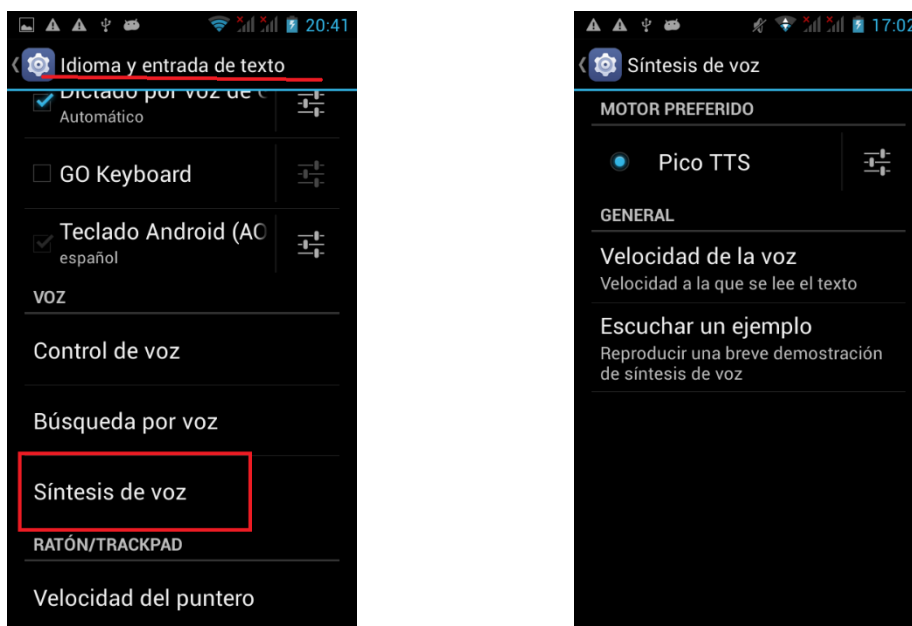


Figura 6. Ajustes de síntesis de voz

2.3.1 Integración de síntesis de voz en la aplicación

Para llevar a cabo la síntesis de voz a texto, la aplicación va a hacer uso del paquete *android.speech.tts* de la API de Android, que permite introducir en una cola el texto que se quiere reproducir. La siguiente tabla [11] muestra las diferentes interfaces y clases que posee el paquete, así como una breve descripción de las mismas.

SynthesisCallback	Una llamada de retorno para regresar los datos de voz sintetizada por un motor de texto a voz.
TextToSpeech.OnInitListener	Definición de una interfaz de una llamada de retorno al ser invocada indicando la finalización de la inicialización del motor <i>TextToSpeech</i> .

(a) Interfaces

SynthesisRequest	Contiene datos requeridos por el motor para realizar la síntesis de texto a voz.
TextToSpeech	Sintetiza una cadena de texto para su reproducción inmediata o para crear un archivo de sonido.
TextToSpeech.Engine	Constantes y nombres de los parámetros para el control de texto a voz.
TextToSpeech.EngineInfo	Información sobre un motor de texto a voz instalado.
TextToSpeechService	Clase base abstracta para implementaciones del motor TTS
UtteranceProgressListener	Escuchador de eventos relacionados con el progreso de un enunciado a través de la cola de síntesis.
Voice	Características y funciones de una voz TTS.

(b) Clases

Tabla 4. Interfaces y clases del paquete *android.speech.tts*

Para integrar esta funcionalidad en la aplicación se requiere de una previa configuración. El primer paso consiste en inicializar el sintetizador de texto a voz mediante la clase *TextToSpeech*, la cual dispone de un método que se lanza en la inicialización del sintetizador y comprueba si el sistema es correcto en el dispositivo. También establece la configuración deseada del sistema, como el idioma, la velocidad o la entonación.

Después de inicializar el sintetizador y configurarlo, pasamos a reproducir el mensaje de texto por voz. Primero se comprueba que la instancia del sistema de síntesis de texto a voz es correcta, que el mensaje que se quiere reproducir es válido y que el sistema no se encuentra ocupado en ese momento. Si es así, se inicia el controlador de la reproducción de mensajes y finalmente se reproduce el mensaje.

Por último, cuando la actividad finaliza se ejecuta el método para desconectar y apagar el sistema de síntesis para dejar los canales de reproducción cerrados y permitir que pueda ser utilizado.

2.4 Aplicaciones sobre residencias

Actualmente en el mercado encontramos diferentes aplicaciones para Android en el ámbito de las residencias de mayores. Estas se han analizado y tenido en cuenta para el desarrollo de la aplicación del proyecto y se detallan a continuación:

- **Residencia Palacios.** Esta aplicación desarrollada por Teldar Capital SL pertenece a un grupo de residencias del mismo nombre de la aplicación situadas en Madrid y en Toledo. La aplicación ofrece al usuario una vez registrado la posibilidad de consultar las actividades y eventos organizadas en las residencias e incluso apuntarse a las mismas. También te permite llamar, chatear y obtener información del centro, acceder a noticias relacionadas y consultar documentos relevantes de los centros [13].

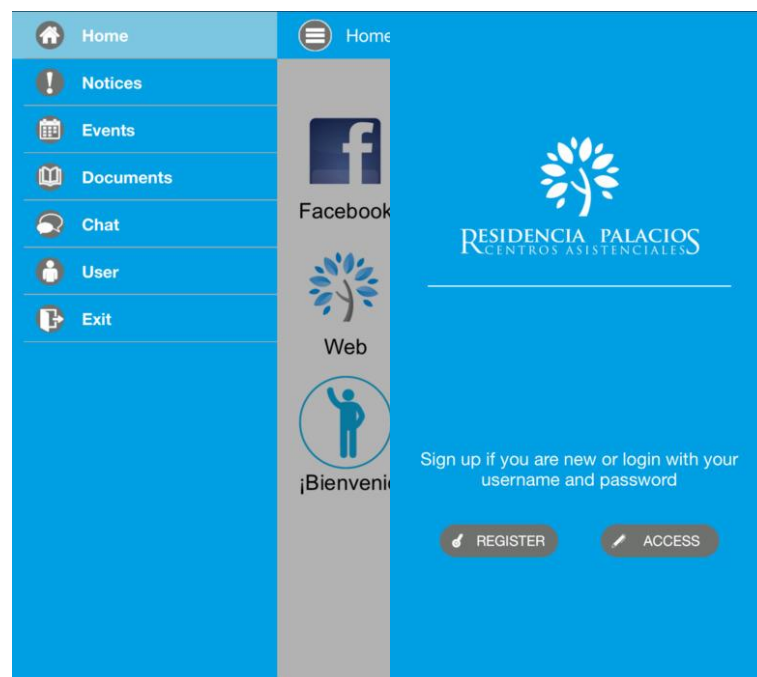


Figura 7. Aplicación Residencia Palacios

- **Residencia de la Soledad.** Esta es la aplicación móvil de la Residencia Ntra. Sra. de la Soledad y del Carmen. En ella podemos encontrar información detallada del centro orientada tanto a familiares como trabajadores del centro. También se puede consultar la agenda, menús diarios, noticias de interés, horarios del centro y los eventos y actividades que organizan. Por otro lado, también es posible contactar con los empleados del centro y acceder a información personal de los pacientes dentro de un área restringida solo para familiares de los mismos y para los trabajadores [14].



Figura 8. Aplicación Residencia de la Soledad.

- **Residencias Sanyres.** Excelia SL desarrollo esta aplicación para las residencias geriátricas del Grupo Sanyres. Esta aplicación fue la primera en salir al mercado en el sector de las residencias geriátricas. En ella se pueden encontrar todos los servicios y programas para ancianos que ofrecen, información sobre promociones para los clientes e información sobre los residentes [15].

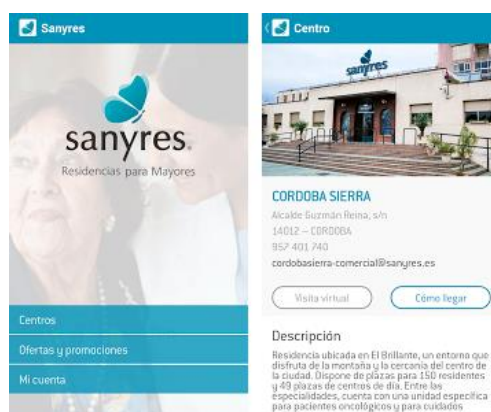


Figura 9. Aplicación Residencias Sanyres

Capítulo 3

Descripción general del sistema

En este capítulo se describe detalladamente el sistema desarrollado y se explican las características fundamentales de cada uno de los módulos, así como la funcionalidad de cada uno. También se analizan las distintas tecnologías utilizadas para la implementación de la aplicación.

3.1 Presentación del sistema

La aplicación para Android desarrollada en este Trabajo de Fin de Grado está destinada a dar soporte a las residencias y centros de día del Grupo Eulen. En concreto, esta aplicación ha sido desarrollada para una en concreto, siendo posible incorporar más en un futuro gracias a que el código de la aplicación está bien separado de los datos específicos de la residencia.

Las diferentes funcionalidades que aborda la aplicación a lo largo de sus diferentes módulos son:

- Consulta de los diferentes centros pertenecientes al Grupo Eulen, así como sus datos contacto y localización.
- Buscador de pacientes, el cual permite buscar los pacientes y acceder a sus datos. Los datos de un determinado paciente sólo pueden consultarse por los familiares y personal del centro con permisos. Esto requerirá de una contraseña única e intransferible para acceder al perfil de un paciente.
- Perfil de un paciente, donde puedes encontrar sus datos médicos, los datos personales del empleado que se encarga de él o enviar un mensaje al centro acerca del paciente.
- Servicios y programas, que muestra la diferente oferta que tienen los centros, así como las actividades que se organizan y ofertan.

Dentro tanto del buscador de pacientes como a la hora de enviar un mensaje al centro, el usuario puede hacerlo mediante la voz. Por otro lado, los datos tanto de los centros como de los pacientes, están almacenadas en una base de datos. En la siguiente figura se muestran las diferentes funcionalidades y se puede ver como se llega de una a otra.

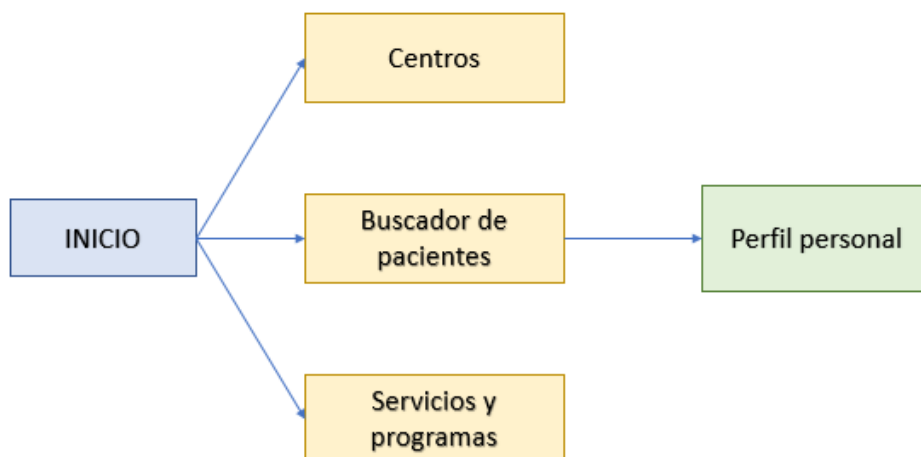


Figura 10. Funcionalidades ofrecidas por la aplicación

3.2 Tecnologías utilizadas

3.2.1 Base de datos

Como se ha comentado anteriormente, la aplicación requiere del uso de bases de datos para almacenar información acerca de los pacientes y de los centros. Para ello se ha utilizado SQLite, el cual es un sistema de gestión de bases de datos y del cual Android proporciona todas las herramientas para utilizarlo. Las razones por las que se ha elegido este sistema son básicamente las siguientes:

- No es necesaria una configuración, es decir, no hay que especificar puertos, tamaños, ubicación de la información...
- No dispone de proceso separado que ejecute un servidor de base de datos, sino que implementa un conjunto de librerías que se encargan de gestionar la base de datos.
- Crea un archivo con el esquema completo de la base de datos, y los datos de este archivo solo pueden ser accedidos por la aplicación y nunca por aplicaciones externas. Esto es importante porque aporta seguridad a los datos.
- Es de código abierto, por lo que facilita a los desarrolladores acceder a la documentación y ejemplos acerca de su uso e implantación.
- Se encuentra embebido en los dispositivos Android, por lo que no es necesaria su descarga, configuración y administración.

La elección de este gestor de bases de datos también cuenta con ciertas limitaciones en comparación con otras como MySQL. En aplicaciones que requieren el uso de bases de datos de gran tamaño o que requieran de alta concurrencia no es aconsejable utilizar SQLite, pero para la realización de este proyecto es suficiente.

SQLite utiliza el lenguaje SQL para realizar las diferentes acciones necesarias sobre las bases de datos dentro de los métodos especificados más adelante. En la siguiente tabla se muestra una clasificación de los diferentes comandos que se pueden utilizar:

Comando	Descripción
CREATE	Crear nuevas tablas, procedimientos almacenados e índices.
DROP	Eliminar tablas, procedimientos almacenados e índices.
ALTER	Modificar tablas agregando campos o combinándolos.
SELECT	Consultar datos de las tablas.
INSERT	Cargar datos en las tablas.
UPDATE	Modificar valores almacenados en campos específicos de las tablas.
DELETE	Eliminar valores de una tabla.

Tabla 5. Resumen de comandos en lenguaje SQL.

Para trabajar con una base de datos SQLite en Android es necesario el uso del paquete android.database de la API de Android. En concreto está el paquete android.database.sqlite, que contiene las clases específicas de SQLite. La siguiente tabla [12] muestra las diferentes interfaces y clases que posee el paquete, así como una breve descripción de las mismas.

SQLiteCursorDriver	Controlador para SQLiteCursors que se utiliza para crearlos y notificarlos eventos.
SQLiteDatabase.CursorFactory	Se utiliza para permitir la devolución de subclases de Cursor cuando se realiza una consulta.
SQLiteTransactionListener	Escuchador de eventos de transacción.

(a) Interfaces

SQLiteClosable	Un objeto creado de SQLiteDatabase que puede ser cerrado.
SQLiteCursor	Implementación de un Cursor que expone los resultados de una consulta de una SQLiteDatabase.
SQLiteDatabase	Expone los métodos para manejar una base de datos SQLite.
SQLiteOpenHelper	Clase de ayuda para gestionar la creación, diseño y actualización de la base de datos.
SQLiteProgram	Clase para compilar programas SQLite.
SQLiteQuery	Representa el resultado de una consulta.
SQLiteQueryBuilder	Clase que ayuda a crear una consulta SQL para enviársela al objeto SQLiteDatabase.
SQLiteStatement	Representar una sentencia que puede ser ejecutada contra una base de datos.

(b) Clases

Tabla 6. Interfaces y clases del paquete android.database.sqlite

De todas estas clases, la que más nos interesa al principio es SQLiteOpenHelper, la cual contiene los métodos necesarios para crear y actualizar una base de datos en Android. Los métodos y una descripción de los mismos se pueden ver en la siguiente tabla:

MÉTODO	DESCRIPCIÓN
close()	Cierra un objeto de tipo base de datos.
getDatabaseName()	Devuelve el nombre de la base de datos abierta.
getReadableDatabase()	Crea y/o abre una base de datos.
getWritableDatabase()	Crea y/o abre una base de datos que será usada para lectura y escritura.
onConfigure(SQLiteDatabase sb)	Configura la conexión a la base de datos.
onCreate(SQLiteDatabase sb)	Se llama cuando se crea la base de datos por primera vez.
onDowngrade(SQLiteDatabase sb, int oldVersion, int newVersion)	Sirve para revertir la versión de la base de datos.
onOpen(SQLiteDatabase sb)	Sirve para abrir la base de datos.

onUpgrade(SQLiteDatabase sb, int oldVersion, int newVersion)	Se llama cuando se quiere aumentar la versión de la base de datos.
---	--

Tabla 7. Métodos de la clase SQLiteOpenHelper.

Después de crear la base de datos, será necesaria su manipulación. Para ello será importante la clase SQLiteDatabase. La siguiente tabla describe los métodos más relevantes que contiene y una descripción de los mismos:

MÉTODO	DESCRIPCIÓN
execSQL(String sql)	Ejecuta una sentencia SQL que no sea un SELECT o cualquier otra que devuelva datos.
insert(String table, String nullColumnHack, ContentValues values)	Sirve para insertar filas en una base de datos.
update(String table, ContentValues values, String whereClause, String[] whereArgs)	Sirve para modificar filas de una base de datos.
delete(String table, String whereClause, String[] whereArgs)	Sirve para eliminar filas de la base de datos.
rawQuery(String sql, String[] selectionArgs)	Ejecuta una consulta SQL (SELECT) y devuelve un objeto de tipo Cursor que apunta a una posición anterior a la primera entrada de la tabla que satisface la consulta.

Tabla 8. Métodos relevantes de la clase SQLiteDatabase

Otros métodos que también pueden ser útiles pertenecen a la clase Cursor y son:

- getCount(): para conocer el número de elementos que retorna una consulta.
- moveToFirst(): para moverse a la primera fila de la tabla.
- moveToNext(): para moverse al siguiente elemento de la tabla.
- isAfterLast(): para saber si hay más elementos en la tabla.

3.2.1.1 Información detallada de la base de datos

La base de datos creada en la aplicación está compuesta por cuatro conjuntos de datos. En las siguientes tablas se pueden ver los atributos que contienen, así como una descripción de cada campo:

Tabla Pacientes

id_paciente	Identificador del paciente en la tabla.
nombre	Nombre del paciente.
apellido	Apellidos del paciente.
edad	Edad del paciente.
nombre_residencia	Nombre de la residencia donde está el paciente.
contraseña	Contraseña del paciente para poder acceder a su perfil personal.
id_trabajador	Identificador del empleado que se encarga de él.
id_datos	Identificador del informe médico del paciente.

Tabla 9. Atributos de la tabla Pacientes

Tabla Datos

id_datos	Identificador del informe médico en la tabla.
enfermedad	Enfermedad del paciente.
fecha_revision	Fecha de la última revisión del paciente.
observaciones	Observaciones médicas del paciente.
fecha_alta	Fecha de incorporación del paciente a la residencia.

Tabla 10. Atributos de la tabla Datos

Tabla Trabajadores

id_trabajador	Identificador del trabajador en la tabla.
nombre	Nombre del trabajador.
apellidos	Apellido del trabajador.
nombre_residencia	Nombre de la residencia donde trabaja.
telefono	Teléfono de contacto del trabajador.

Tabla 11. Atributos de la tabla Trabajadores

Tabla Residencias

nombre	Nombre de la residencia.
calle	Calle donde se encuentra la residencia.
población	Población donde se encuentra la residencia.
email	Email de contacto de la residencia.
teléfono	Teléfono de contacto de la residencia

Tabla 12. Atributos de la tabla Residencias

En el siguiente diagrama se muestra el modelo de la base de datos, en el cual se indica cómo se compone cada tabla, las características de cada atributo y las relaciones de unas tablas con otras.

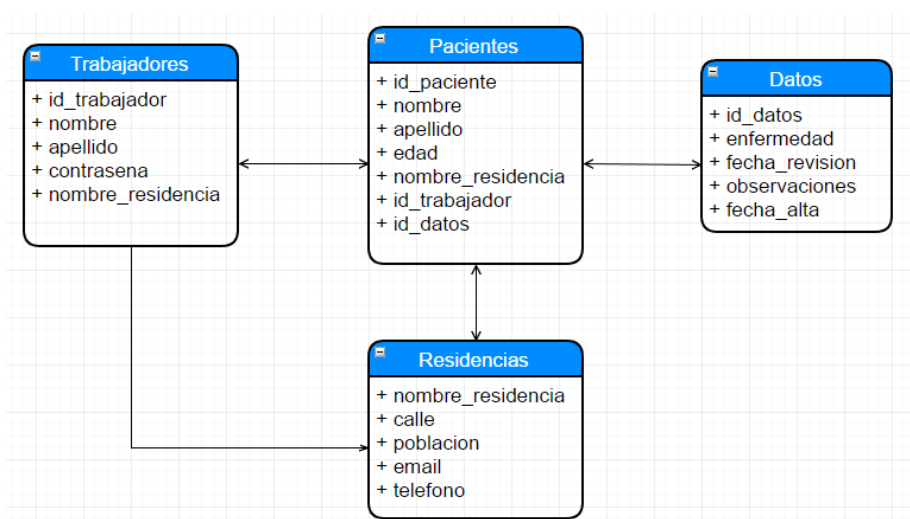


Figura 11. Diagrama del modelo de base de datos

3.3 Módulos del sistema

Un módulo del sistema es cada uno de los bloques que aporta un conjunto de funcionalidades relacionadas con los objetivos del proyecto. Estos módulos son manejados por el usuario y pueden relacionarse entre sí. A continuación, se describen brevemente los diferentes módulos del sistema, los cuáles se detallarán en el siguiente capítulo:

- Módulo de Inicio. Este módulo cubre la gestión de la información principal que se muestra en la primera pantalla de la aplicación, como son las diferentes opciones para navegar por ella.
- Módulo de Servicios. Este módulo cubre la gestión de la información acerca de los servicios y programas ofrecidos.
- Módulo de Centros. Este módulo cubre la gestión de todos los centros, su información y su método de contacto.
- Módulo de Pacientes. Este módulo cubre la gestión de todos los pacientes registrados que el usuario puede buscar dentro de la aplicación. Comprende un conjunto de submódulos relacionados con la información personal de cada paciente.

Capítulo 4

Descripción detallada de los componentes del sistema

4.1 Módulo “Inicio”

En su primera interacción con la aplicación, al usuario le aparecerá la pantalla principal mostrada en la Figura 9. En esta pantalla se puede ver el logo de la empresa Grupo Eulen, para la cuál se ha desarrollado la aplicación, y tres diferentes botones. Cada uno de ellos lleva a un módulo diferente: el botón “Centros” nos lleva al módulo de Residencias, el botón “Servicios y programas” nos lleva al módulo de Servicios y el botón “Buscador de pacientes” nos lleva al módulo de Pacientes.



Figura 12. Módulo “Inicio” de la aplicación.

4.2 Módulo “Servicios”

Cuando el usuario pulsa el botón de “Servicios y programas” en el módulo inicial, al usuario se le muestran las diferentes opciones que se ofrecen. Esto se puede ver en las Figuras 10 y 11.

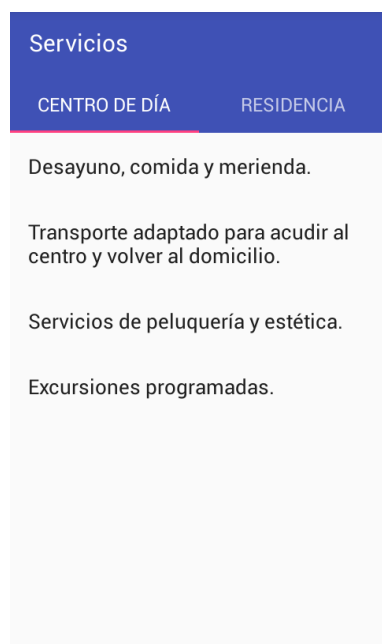


Figura 13. Pantalla 1 del módulo “Servicios”

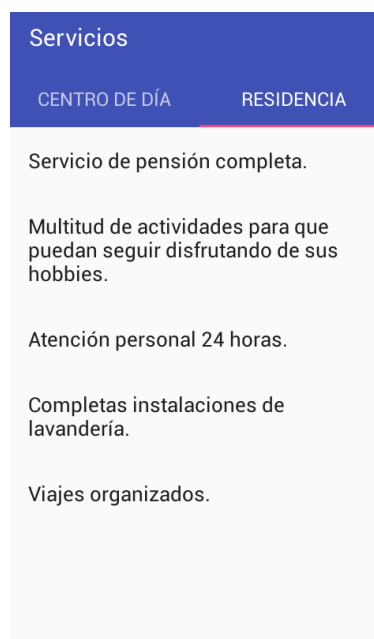


Figura 14. Pantalla 2 del módulo “Servicios”

La pantalla de este módulo está formada por un sistema de pestañas con dos pestañas, debido a que el Grupo ofrece dos tipos de servicios. En la primera pestaña podemos ver las diferentes opciones que se ofrecen como centro de día. En la segunda pestaña podemos ver los servicios que ofrece como residencia. Si queremos saber los servicios concretos de un centro lo mejor es contactar directamente con ella accediendo a sus datos en el siguiente módulo.

4.3 Módulo “Centros”

Una vez el usuario a pulsado en el botón “Centros” de la pantalla inicial, el usuario es redirigido a la pantalla mostrada en la Figura 12.



Figura 15. Pantalla principal del módulo “Centros”

En la pantalla se puede ver un listado de todas las residencias que contiene la base de datos. Al iniciar la actividad, la base de datos crea la tabla “Residencias” e introduce los datos mediante la siguiente consulta:

```
public long guardarResidencia(SQLiteDatabase sqLiteDatabase, Residencia residencia){  
    return sqLiteDatabase.insert(  
        ResidenciaEntry.TABLA_RESIDENCIAS,  
        null,  
        residencia.toContentValues());  
}
```

Figura 16. Ejemplo de inserción en Base de Datos.

Este método para guardar datos en la tabla recibe como parámetros la base de datos creada en un método anterior y un objeto de tipo Residencia, el cual contiene todos los atributos que se han introducido como columnas de la tabla Residencias. También podemos ver que el tercer parámetro que recibe el método *insert* es el objeto de tipo Residencia llamando al método *toContentValues()*. Este método que se encuentra en la clase Residencia se encarga de transformar los parámetros del objeto en valores de tipo ContentValues, los cuales son requeridos para insertar datos en la base de datos. Este procedimiento se ha seguido para la inserción de todos los datos en la base de datos, llamada “GrupoEulen.db”.

4.3.1 Submódulo “Detalles de la residencia”

Partiendo de la pantalla inicial del módulo de centros, el usuario puede consultar los datos detallados de una residencia. Para ello basta con pulsar en cualquiera de los elementos de la lista. En la Figura 14 se puede ver el resultado de esta acción.

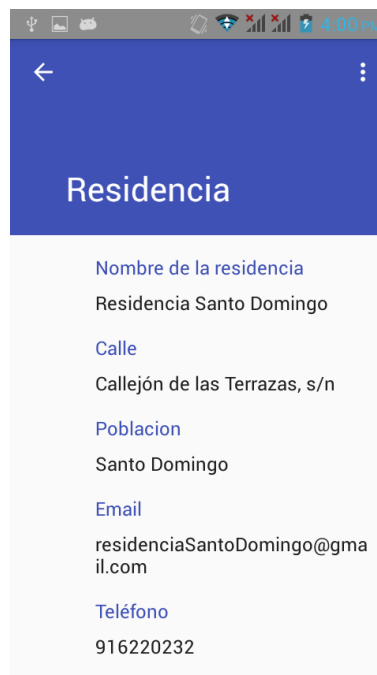


Figura 17. Submódulo “Detalles de la residencia”

En la figura podemos ver los datos de la residencia seleccionada, en este caso de la “Residencia Santo Domingo”. Se puede observar la calle y la población dónde está ubicada, así como detalles de contacto como el email o el teléfono. También cabe la posibilidad de volver atrás a la lista de residencias mediante la flecha situada en la parte superior izquierda de la pantalla.

Para que la obtención de los datos de la residencia fuera posible se ha realizado la siguiente consulta de la base de datos:

```
public Cursor getResidenciaById(String residenciaId) {  
    Cursor c = getReadableDatabase().query(ResidenciaEntry.TABLA_RESIDENCIAS, null,  
        ResidenciaEntry.ID + " LIKE ?", new String[]{residenciaId}, null, null, null);  
    return c;  
}
```

Figura 18. Ejemplo de consulta en base de datos.

En la consulta se accede a la tabla Residencias para obtener todos los datos de la residencia cuyo id coincida con el recibido como parámetro. Ese id se obtiene porque al pulsar la residencia en la lista, el id es pasado mediante el objeto intent con una clave que le referencia, para que luego la nueva actividad pueda recogerlo mediante la misma clave. Se podrá ver un ejemplo de ello en el módulo de Pacientes, el cuál utiliza esto de manera más compleja.

4.4 Módulo “Pacientes”

Cuando el usuario pulsa el botón “Buscador de pacientes” de la pantalla principal del módulo de inicio, la aplicación muestra la siguiente pantalla.

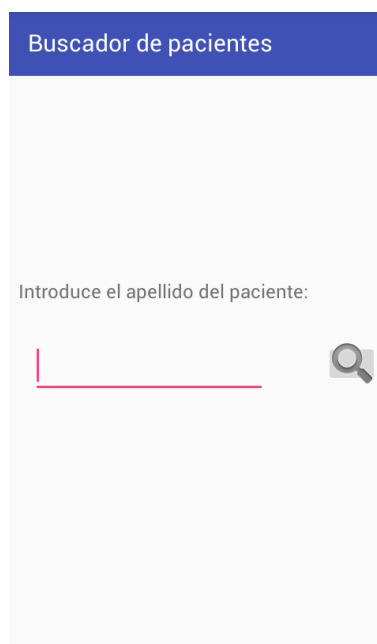


Figura 19. Pantalla inicial del módulo de pacientes.

En la pantalla aparece un cuadro de texto a rellenar y una lupa. También podemos ver como al usuario se le indica que introduzca el apellido del paciente para realizar la búsqueda. Al igual que en el módulo “Residencias”, al iniciar la actividad se ha creado la tabla Pacientes y se han insertado los datos de los pacientes. En la Figura 17 se puede observar como el usuario puede introducir los apellidos tanto por voz como por escrito.

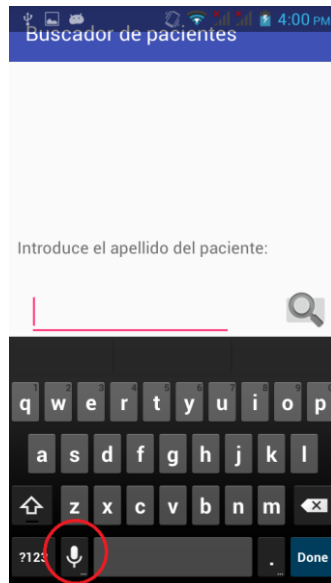


Figura 20. Ejemplo de entrada por voz de la aplicación.

Una vez el usuario ha introducido los apellidos en el buscador (Figura 18), debe pulsar la lupa para que se realice la búsqueda en la base de datos. La actividad captura el valor del apellido introducido y lo mete como parámetro en la consulta a la base de datos. El valor es capturado mediante:

```
apellidos = (EditText) root.findViewById(R.id.editText);
String apellidosPaciente = apellidos.getText().toString();
```

El valor “apellidosPacientes” es el que se introduce en la consulta al igual que se hace en el módulo “Residencias” con el id.

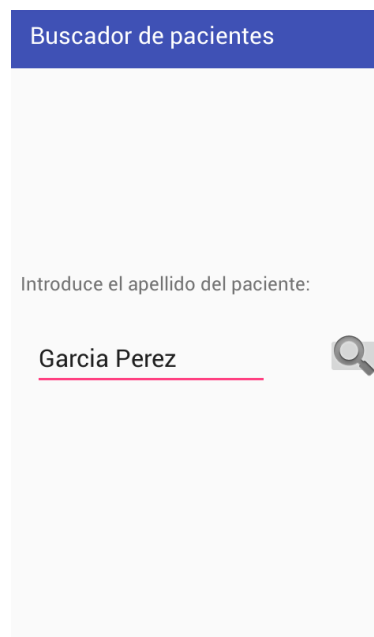


Figura 21. Ejemplo de búsqueda de un paciente.

4.4.1 Submódulo “Detalles del paciente”

Una vez se ha consultado en la base de datos, se obtienen los datos asociados al paciente con el apellido indicado y se muestran.

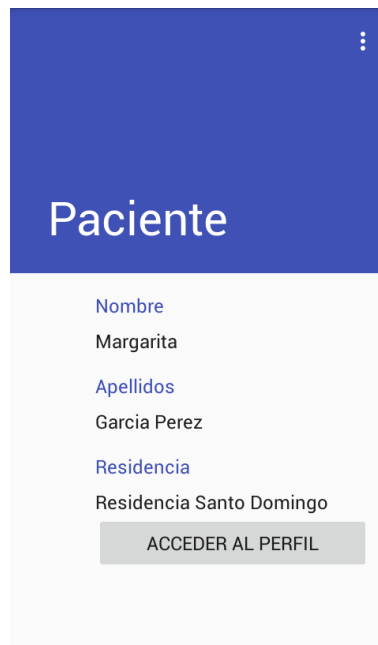


Figura 22. Pantalla del submódulo “Detalles del paciente”.

Como detalles del paciente podemos ver los más básicos, como son el nombre, los apellidos y la residencia en la que se encuentra. Para poder ver más datos del paciente, es necesario pulsar el botón “Acceder al perfil”.

4.4.2 Submódulo “Login”

Una vez el usuario ha pulsado en el botón, aparece la siguiente pantalla en la aplicación.



Figura 23. Pantalla del submódulo “Login”.

En la pantalla aparece un texto que indica que debes meter la contraseña del paciente. A la hora de crear la tabla Pacientes, se ha introducido una columna de contraseña (ver Figura 21).

```
sqliteDatabase.execSQL("CREATE TABLE " + PacienteEntry.TABLA_PACIENTES + " ("
    + PacienteEntry._ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
    + PacienteEntry.ID + " TEXT NOT NULL,"
    + PacienteEntry.NOMBRE + " TEXT NOT NULL,"
    + PacienteEntry.APELLIDOS + " TEXT NOT NULL,"
    + PacienteEntry.EDAD + " TEXT NOT NULL,"
    + PacienteEntry.CONTRASENA + " TEXT NOT NULL,"
    + PacienteEntry.RESIDENCIA + " TEXT NOT NULL,"
    + PacienteEntry.TRABAJADOR + " TEXT,"
    + PacienteEntry.DATOS + " TEXT NOT NULL,"
    + "UNIQUE (" + PacienteEntry.ID + ")");
```

Figura 24. Ejemplo de creación de la tabla Pacientes.

En esta figura se pueden ver todos los atributos que posee un paciente. La columna “contrasena” se ha introducido por temas de privacidad. Esta contraseña solo es conocida por los familiares del paciente y por el trabajador que se encarga de él. Cuando el paciente es dado de alta en la aplicación, se les proporciona la contraseña a ambas partes interesadas. Esto es por temas de protección de datos con los cuales hay que tener mucho cuidado, ya que el perfil del paciente contiene datos personales confidenciales para la entidad y los familiares. En la Figura 22 se puede ver que es lo que pasa cuando se introduce una contraseña incorrecta.

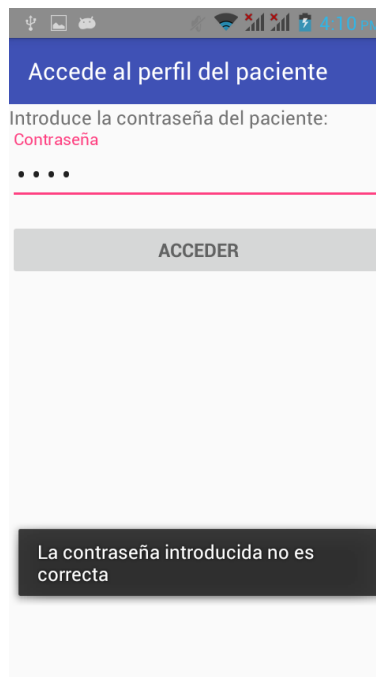


Figura 25. Ejemplo de contraseña incorrecta.

4.4.3 Submódulo “Perfil del paciente”

Si el usuario ha introducido la contraseña correcta, será redirigido a la pantalla principal de este submódulo.

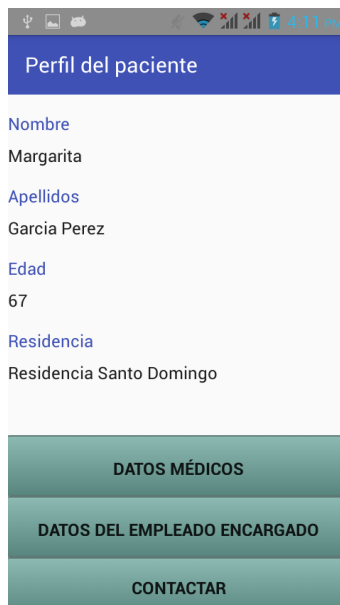


Figura 26. Pantalla principal del submódulo “Perfil del paciente”.

En esta pantalla se pueden ver los datos generales del paciente, como son nombre, apellidos, edad y residencia donde se encuentra. Estos datos son los mismos que se mostraban en la pantalla principal del submódulo “Detalles del paciente” a excepción de la edad, que sólo se muestra en esta. Aquí el usuario, que es o un familiar o el trabajador asociado a este paciente, tiene la opción de consultar los datos médicos, consultar los datos del empleado encargado o, si es un familiar, enviar un mensaje al trabajador.

Si el usuario decide consultar los datos médicos del paciente, una vez pulsado el botón, le aparecerá la pantalla mostrada en la Figura 24.



Figura 27. Pantalla de datos médicos del paciente.

Esta pantalla muestra los datos personales sanitarios del paciente. Se puede ver la enfermedad del paciente, la fecha de su última revisión, la fecha de ingreso en el centro y las observaciones a tener en cuenta por parte del personal del centro. Esto se puede obtener debido a que al realizar la consulta en la base de datos cuando se busca el paciente, los datos obtenidos persisten entre actividades, y una vez llegado a este punto se realiza una búsqueda en la tabla Datos por el id de los datos del paciente, el cual es un identificador único.

Si el usuario decide consultar los datos personales del empleado encargado del paciente, una vez pulsado el botón correspondiente, le aparecerá la siguiente pantalla.

The screenshot shows a mobile application interface with a blue header containing the word 'Trabajador' in white. Below the header, on a light gray background, are four labels in blue text: 'Nombre', 'Apellidos', 'Residencia', and 'Telefono'. Each label is followed by its corresponding value in black text: 'Rosa', 'Calvente Menendez', 'Residencia Santo Domingo', and '65432134'.

Nombre	Rosa
Apellidos	Calvente Menendez
Residencia	Residencia Santo Domingo
Telefono	65432134

Figura 28. Pantalla de datos personales del trabajador.

En esta pantalla se pueden observar los datos personales del trabajador. Se pueden ver su nombre, apellidos, residencia donde trabaja y el teléfono de contacto. Esto es útil puesto que los familiares pueden contactar directamente con el trabajador encargado del paciente si surge alguna urgencia y necesita contactar con él.

Por último, tenemos la opción de contactar con el trabajador mediante el tercer botón de la pantalla principal del submódulo “Perfil del paciente”. Si el usuario pulsa el botón, le aparecerá un cuadro de texto donde puede escribir el mensaje (tanto por voz como mediante el teclado) y al pulsar el botón “*Enviar*” el mensaje será enviado al trabajador.

The screenshot shows a mobile application interface for contacting a worker. It has a blue header with a back arrow and the text 'Contactar con el emplea...'. Below the header is a text input field containing the message 'Hola buenas tardes queria hablar con usted esta tarde'. A red underline is visible under the text. Below the input field is a gray button labeled 'ENVIAR'. At the bottom of the screen, there is a black notification box with the text 'Mensaje enviado al trabajador' in white.

Figura 29. Pantalla para contactar con el trabajador.

Capítulo 5

Evaluación de la aplicación

5.1 Método de evaluación

Con el propósito de optimizar los resultados de uso de la aplicación, será útil realizar una evaluación de la aplicación, estudiando así su usabilidad y su rendimiento. La ayuda de todos los que han probado la aplicación ha sido importante para completar las conclusiones y trabajo futuro. Se han encontrado pequeños fallos que se comentaran en el capítulo 6.

A parte de utilizar la aplicación, los usuarios han completado una encuesta diseñada en Google Forms que recoge las opiniones subjetivas y grado de satisfacción que han tenido navegando por la aplicación. La elección de esta plataforma para diseñar el cuestionario es debida a que es una herramienta gratuita proporcionada por Google y es fácil distribuir los enlaces de la encuesta a los usuarios y recoger los resultados de la misma.

Encuesta de evaluación de la aplicación de e-health

Indique la dificultad de manejo táctil de la aplicación

- ☐ Muy fácil
- ☐ Fácil
- ☐ Normal
- ☐ Difícil
- ☐ Muy difícil

¿Le ha sido fácil la navegabilidad por la aplicación?

- ☐ No
- ☐ Si
- ☐ Si, pero ha sido difícil
- ☐ Si, ha sido muy sencillo

Evalúe la velocidad de la interacción

12345

Lenta

☐☐☐☐☐

Rápida

Evalúe la calidad de la información obtenida

12345

Mala

☐☐☐☐☐

Buena

¿Cómo ha sido la interacción por voz con la aplicación?

☐ Mal, me ha resultado imposible

☐ Mal, ha sido difícil

☐ Normal, no lo he usado

☐ Bien, era claro y sencillo.

☐ Bien, ha sido de gran utilidad

¿Se ha producido algún error mientras utilizaba la aplicación?

☐ No

☐ Si (indique el error)

☐ Otro:

Considera que la aplicación es útil para residencias de mayores

☐ Si, la veo indispensable

☐ Si, puede ser útil

☐ Normal.

☐ No, no le veo la utilidad

☐ No, me parece absurda

ENVIAR

Figura 30. Encuesta de evaluación de la aplicación.

5.2 Resultados de la evaluación

La encuesta de evaluación ha sido completada por 13 personas, los cuales utilizaron la aplicación libremente, eligiendo por donde navegar y accediendo a los módulos que querían para probar la aplicación al completo. La encuesta ha sido realizada generalmente por familiares y amigos, debido a que no ha sido posible acudir a ningún centro para que probaran la aplicación. Las respuestas han sido imparciales y aparte los usuarios han dado consejos de mejora que se comentaran en el capítulo 6 como trabajo futuro.

Los resultados de las encuestas se han obtenido gracias a la herramienta Google Forms utilizada para diseñarla, puesto que ofrecía un apartado donde se resumían las diferentes respuestas obtenidas tanto de manera individual como de manera general. En las siguientes figuras se muestran los resultados generales obtenidos con las 13 respuestas.



Figura 31. Resultados de la evaluación: pregunta 1.

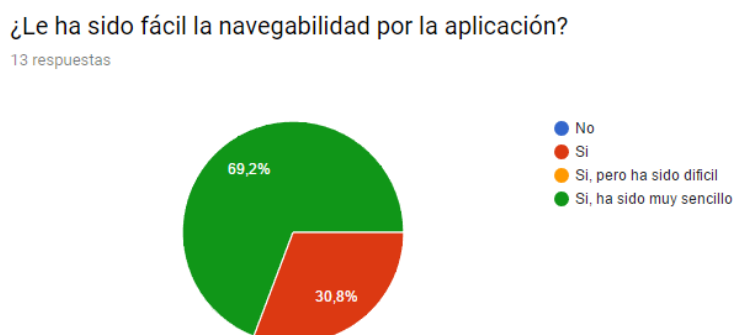


Figura 32. Resultados de la evaluación: pregunta 2.

Evalúe la velocidad de la interacción

13 respuestas

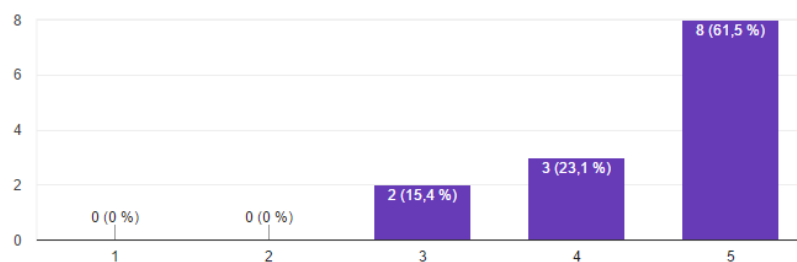


Figura 33. Resultados de la evaluación: pregunta 3.

Evalúe la calidad de la información obtenida

13 respuestas

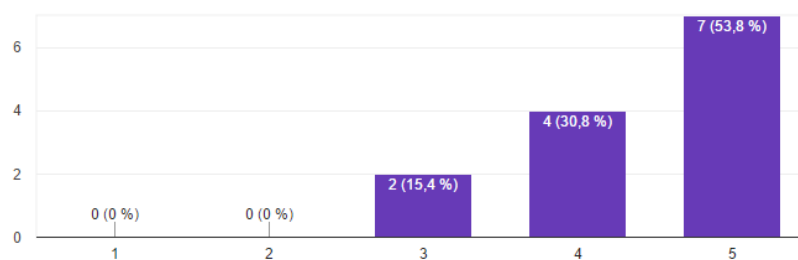


Figura 34. Resultados de la evaluación: pregunta 4.

¿Cómo ha sido la interacción por voz con la aplicación?

13 respuestas

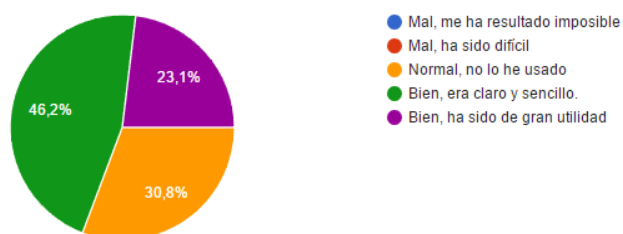


Figura 35. Resultados de la evaluación: pregunta 5.

¿Se ha producido algún error mientras utilizaba la aplicación?

13 respuestas

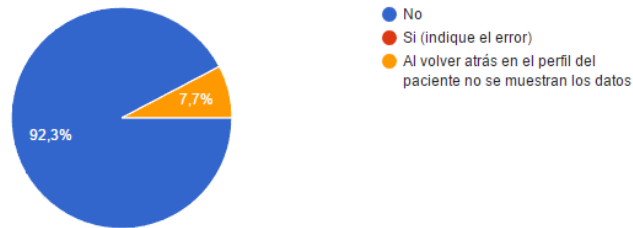


Figura 36. Resultados de la evaluación: pregunta 6.

Considera que la aplicación es útil para residencias de mayores

13 respuestas

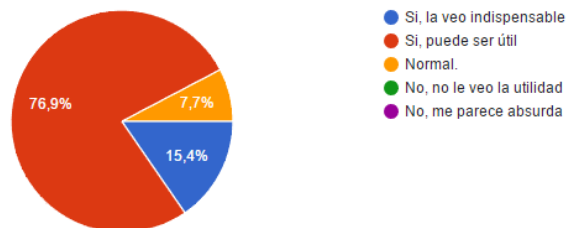


Figura 37. Resultados de la evaluación: pregunta 7.

Una vez obtenidos los resultados, se analizan para poder sacar conclusiones acerca de ellos:

- Por lo general, el manejo táctil de la aplicación y la navegabilidad por la misma ha sido sencillo y de valoración positiva. En ningún caso a un usuario le ha causado ningún problema debido a que no hay ninguna respuesta negativa acerca de estas preguntas.
- La aplicación tiene una velocidad de interacción buena, en ningún caso ha causado ningún problema.

- La calidad de la información obtenida es bastante buena, en ningún caso ha recibido puntuación negativa. Si es cierto que poco más del 50% de los encuestados le ha dado el valor más alto (5), por lo que podemos deducir que la información ofrecida por la aplicación requiere una mejora.
- La interacción por voz es un aspecto valorado positivamente, era fácil saber cuándo y cómo usarlo. También es importante el número de personas que no lo han usado.
- De todos los usuarios encuestados, solo se ha encontrado un error que se refleja en el gráfico de la Figura 36.
- La mayoría de usuarios les parece útil la aplicación, resultado muy positivo. Solo un porcentaje muy pequeño considera a aplicación como normal. Pocos usuarios han valorado la aplicación como indispensable, por lo que podemos decir que necesita mejoras para que este porcentaje aumente.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusiones

La aplicación desarrollada consiste en un gestor de residencias para dar soporte a las residencias del Grupo Eulen. La aplicación permite consultar diversa información acerca de los centros, como puede ser información de localización y contacto de los centros, servicios y programas que ofrecen, y detalles de los pacientes que están en la base de datos de la empresa. Estas funcionalidades cubren las propuestas que se determinaron a en el momento de la planificación y el diseño de la aplicación.

La aplicación cuenta con un sistema multimodal, es decir, el usuario tiene la posibilidad de utilizar como método de entrada tanto la voz como el teclado. Esta característica ha sido implementada para un conjunto muy reducido de funciones, pero la aplicación no lo requería en ningún otro apartado. Resultaba interesante aplicar el sistema de control por voz a la aplicación debido a que aporta facilidades al usuario para el manejo de cualquier aplicación, puesto que, como se explica en el capítulo 2, puede utilizarlo en múltiples situaciones y sin la necesidad de tener acceso a internet.

Todos los datos que se pueden consultar en la aplicación están almacenados en una base de datos SQLite, puesto que esta herramienta está incorporada en Android y contiene todas las funciones necesarias para el almacenamiento y manejo de datos. Este gestor de bases de datos no necesita configuración previa, por lo que para utilizarlo solo ha sido necesario conocer bien las interfaces y clases que ofrecía el API de Android para desenvolverse en el manejo de la herramienta. Gracias al diseño relacional realizado sobre la base de datos previo a la implementación de la misma, ha sido posible adaptar con mayor facilidad los datos necesarios de la aplicación y definir las consultas sobre la misma para que estas fueran óptimas. Como la aplicación no requería de una gran cantidad de datos, este gestor ha sido suficiente para almacenar las cuatro tablas con sus diferentes datos. Para que las operaciones sobre la base de datos no perjudicaran el funcionamiento ni la velocidad de la aplicación, estas se han ejecutado en un hilo diferente de la aplicación como una tarea asíncrona.

Gracias al desarrollo de la aplicación ha sido posible adquirir muchos conocimientos acerca de la arquitectura y características del sistema operativo Android, lo cual antes era un mundo totalmente desconocido. En los últimos tiempos el tema de desarrollar aplicaciones está en auge, por lo que no ha sido muy complicado encontrar ayuda en

internet para desenvolverse y avanzar en la implementación de la aplicación. A parte de eso, Android dispone de manera pública y gratuita la información sobre todos los paquetes disponibles, sus interfaces y sus clases, así como ejemplos acerca de cómo utilizarlos. Que el desarrollo se realice mediante código Java y XML ha influido positivamente puesto que se pueden encontrar cantidad de ejemplos de código en Internet. También cabe destacar que el lenguaje de programación Java es uno de los más estudiados en la carrera, por lo que esto no ha supuesto ningún problema a la hora de programar en el Android Studio.

Partiendo de la evaluación obtenida por parte de diferentes usuarios, se concluye con una valoración positiva de la aplicación. Los usuarios la consideran como usable, intuitiva y fácil de manejar. Es valorada como una aplicación útil, por lo que su implementación en los centros puede tener como respuesta que se cumplan los objetivos planteados al comienzo del proyecto. El objetivo principal que se pretende conseguir con la implementación de la aplicación es el de optimizar el nivel de vida en la vejez, concepto denominado como envejecimiento activo.

6.2 Trabajo futuro

A continuación, y para concluir, se detallan los aspectos a trabajar en el futuro para mejorar las prestaciones de la aplicación desarrollada. Esto incluye mejoras en los módulos implementados, así como nuevas funcionalidades que se pueden incorporar para mejorar la satisfacción del usuario.

- En el módulo de las residencias, más concretamente en el detalle de las mismas, añadir un enlace con Google Maps que permita mostrar la ubicación de la residencia, así como una ruta desde la ubicación seleccionada.
- En un principio, la aplicación fue pensada para facilitar la movilidad en el interior de la residencia. Esto no fue posible puesto que se necesitaba una autorización que nunca llegó para poder visitar una residencia y obtener los planos de la misma. Como trabajo futuro se plantea la posibilidad de incorporar planos internos de las residencias. Esto permitirá a los pacientes convertirse en usuarios de la aplicación, permitiéndoles moverse por la residencia con mayor soltura. La idea era que la aplicación pudiera guiar al usuario por la residencia mediante la voz para mejorar aún más la funcionalidad.

- Dentro de los detalles de la residencia, incorporar un botón que permita al usuario contactar con la residencia instantáneamente. Mediante el botón se abriría la aplicación del teléfono en el caso de pulsar el teléfono o la aplicación de correo en el caso del email de contacto. Esto sería útil también en los detalles del trabajador a la hora de contactar por teléfono con él.
- Añadir información a los datos del centro, como capacidad, tipos de habitaciones o actividades más concretas que se realicen dentro del mismo.
- En el apartado de contactar con el trabajador, establecer un chat en vez de un cuadro de texto para poder mantener una conversación en vez de enviar un único mensaje.

Glosario

E-health – Sanidad electrónica (*Electronic Health*)

M-health – Sanidad en dispositivos móviles (*Mobile-Health*)

SDK – Kit de desarrollo de software (*Software Development Kit*)

JDK – Kit de desarrollo de Java (*Java Development Kit*)

TTS –Conversor texto-voz (*Text-To-Speech*)

Bibliografía

[1] Ureña, A. P. (2016). *"Las TIC en los hogares españoles"*, en *LA SOCIEDAD EN RED Informe Anual 2015*. Madrid: Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información.

[2] Clarke, M. &. (2014). *Tendencias en las tecnologías móviles y sus aplicaciones*. Madrid: Fundación EOI.

[3] Adibi, S. (2015). *Mobile Health. A technology Road Map*. Springer.

[4] Envejecimientoactivo2012.net. (2017). *2012 Año Europeo del Envejecimiento Activo*. [en línea] Disponible en: <http://www.envejecimientoactivo2012.net/Menu28.aspx>

[5] Jia-Ni Lu (2012). *Human-machine Interaction Based on Voice*. ELSEVIER. Disponible en: <http://www.sciencedirect.com/science/article/pii/S221267161200251X>

[6] Griol, D. (2007). *Desarrollo y Evaluación de Diferentes Metodologías para la Gestión Automática del Diálogo*. Tesis doctoral. Valencia: Universidad Politécnica de Valencia [en línea]. Disponible en: <http://users.dsic.upv.es/~dgriol/papers/TesisDgriol.pdf>

[7] LListerri, J. (2006). Introducción a los sistemas de diálogo en J. Llisterri & M. J. Machuca (Eds.), *Los sistemas de diálogo*. (pp. 11-21).Bellaterra – Soria: Universitat Antònoma de Barcelona – Fundación Duques de Soria [en línea]. Disponible en: http://liceu.uab.cat/~joaquin/publicacions/Llisterri_06_Sistemas_Dialogo.pdf

[10] Developer.android.com. (n.d.). *android.speech / Android Developers*. [en línea] Disponible en: <https://developer.android.com/reference/android/speech/package-summary.html>

[11] Developer.android.com. (n.d.). *android.speech.tts / Android Developers*. [en línea] Disponible en: <https://developer.android.com/reference/android/speech/tts/package-summary.html>

[12] Developer.android.com. (n.d.). *android.database.sqlite / Android Developers*. [en línea] Disponible en: <https://developer.android.com/reference/android/database/sqlite/package-summary.html>

[13] Play.google.com. (2017). *Residencia Palacios*. [en línea] Disponible en:
<https://play.google.com/store/apps/details?id=com.teldarcapital.eventelling.residenciapalacios&hl=es> 419

[14] Play.google.com. (2015). *Residencia de la Soledad*. [en línea] Disponible en:
<https://play.google.com/store/apps/details?id=app83360.vinebre&hl=es>

[15] Play.google.com. (2015). *Residencia Sanyres*. [en línea] Disponible en:
<https://play.google.com/store/apps/details?id=es.sanyres.app&hl=es>

Anexo final

DECLARACION DE ORIGINALIDAD

Yo, Manuel Estada Calvente declaro que el TFG *"Desarrollo de una aplicación móvil dentro del campo del m-health"* es totalmente original mío, que no ha sido presentado en ninguna otra universidad como TFG y que todas las fuentes que han sido utilizadas han sido adecuadamente citadas y aparecen en las referencias bibliográficas.

Colmenarejo, a 21 de junio de 2017

Firma: